Coordinated Science Campaign Scheduling for Sensor Webs

Robert Morris[1]
Jennifer Dungan[1]
Will Edgington[2]
1. NASA Ames Research Center
2. QSS Group Inc
Email: morris@email.arc.nasa.gov

Jenny Williams[3]
Jean Carlson[3]
Damian Fleming [3]
and Terri Wood[4]
3. Computer Sciences Corporation
4. NASA Goddard Space Flight Center

***Abstract*— Future Earth observing missions will study different aspects and interacting pieces of the Earth's eco-system. Scientists are designing increasingly complex, interdisciplinary campaigns to exploit the diverse capabilities of multiple Earth sensing assets. In addition, spacecraft platforms are being configured into clusters, trains, or other distributed organizations in order to improve either the quality or the coverage of observations. These simultaneous advances in the design of science campaigns and in the missions that will provide the sensing resources to support them offer new challenges in the coordination of data and operations that are not addressed by current practice. For example, the scheduling of scientific observations for satellites in low Earth orbit is currently conducted independently by each mission operations center. An absence of an information infrastructure to enable the scheduling of coordinated observations makes it difficult to execute campaigns involving multiple assets. This paper proposes a software architecture and describes a prototype system called DESOPS (Distributed Earth Science Observation Planning and Scheduling) that will address this deficiency.**

I. Introduction

NASA's Earth science vision emphasizes the importance of establishing a tighter link among Earth science models, data analysis, and observational activities at all relevant spatial and temporal scales. To enable such a tight linkage, there needs to be an associated information infrastructure binding the cycle of observation, on-board data handling and computing, transmission to ground, storage, data mining and product distribution to support activities such as inverse modeling, data assimilation and model evaluation. The cyclical nature of the linkage implies that new observation goals will emerge out of the products generated from previous observations.

The envisioned future remote sensing environment consists of large numbers of networked sensors that are frequency-agile and capable of multi-scene observations from different space vantage points. Data acquired from such platforms will be merged with those acquired by more traditional systematic missions (such as Landsat). Second, for the purpose of validation and model robustness, data acquired by other observational platforms, including sub-orbital measurements using ground-, airborne-, and balloon sensors, will be merged with data from remote sensing platforms to form a sensor-web. Furthermore, the focus will be on the development of complex compositional Earth science models, wherein focused process models combine iteratively to form interactive multi-component models that simulate the coupled behavior of two or more Earth system components. A complete multi-component model of the Earth is considered the holy grail of Earth science research.

Consequently, Earth scientists will require data from multiple sources distributed in space, over significant periods of time, with choices available to the users of the data with respect to when, where and how these data will be acquired. Planning and executing a series of observations will benefit from information technology that provides an interface to the sensing resources available to meet observation goals, in a way analogous to the way that web-based archive data retrieval tools such as GLOVIS (http://glovis.usgs.gov) provide an interface for retrieving data that has been acquired in the past.

This paper provides an overview of a set of capabilities for addressing the need for coordination of observations. The system is based on a methodology called model-based observing. By *model-based observing* is meant here the process of allocating and scheduling sensing resources based on the goal of validating a specific hypothesis derived from an Earth science model. Model-based observing allows observation scheduling to be *campaign-driven*, where a campaign is defined as a systematic set of activities undertaken to meet a particular science objective. Campaign goals require the collection of data on several variables, on different observing resources at different times and potentially at varying locations.

In the next section we present the overall architecture for model-based observing that links the Earth science community to observation resources. Part of the architecture forms the set of capabilities for coordinating observations, which is the focus of the remainder of the paper. These capabilities are organized into a set of components of a system, called DESOPS (Distributed Earth Science Operations Planning and Scheduling System).

II. Architecture for Model-based Observing

Model-based observing requires coordinating the assignments of observation tasks among a collection of remote sensors or sub-orbital platforms such as ground-, airborne-, and balloon sensors, possibly configured into an organization (e,g, a train or a sensor web) [4]. It is assumed here that each sensing or satellite resource has a distinct, geographically separated, operations team for managing the daily activities of each sensor. Using an economic metaphor, the interests and objectives of these "resource owners" are different from those of the consumers; in particular, the users want maximum utility of the data received associated with their specific science goals, whereas the resource owners have other, potentially conflicting goals. In this regard, the operation environment for model-based observing offer challenges similar to those potentially solved by so-called *computational grid* systems [9], namely the need to provide visibility and access to a set of resources while maintaining the security and autonomy of operations for each.

A system for coordinating observations provides an added layer between the individual scientist and mission operations planning. The coordination layer consists of tools that allow consumers and providers to express requirements for facilitating the successful completion of observation goals. Resource users need to specify a set of measurements as well as a utility model for the data to be acquired. They need to be able to specify constraints on cost and completion time for their campaign goals. They need a mechanism to act as a broker to identify available resources and dynamically submit requests to schedule observations on them. The tool should monitor the execution of these requests and adapt to uncertainties in the availability of resources during execution, which potentially involves rescheduling observations on the same or different resources. Resource owners need a flexible means to specify constraints on the utilization of the resource, as well as a way to continuously supply updated statistics on current load and capacity. They need a system that will facilitate improved utilization of their resource without interrupting normal mission operations.

The overall architecture is displayed in Figure 1. The coordination layer is labeled DESOPS (Distributed Earth Science Observation Planning and Scheduling). DESOPS consists of the information infrastructure for constructing campaign plans involving a collection of sensors and enables more direct contact between Earth Scientists and the mission planning process. The next sections describe these capabilities in more detail.[1]

III. DESOPS Capabilities

The DESOPS core function is to generate and execute Earth science campaign plans. Campaign plan generation includes managing a set of user-specified constraints on feasible plans, employing a set of optimization criteria for ordering feasible plans based on user preferences and utilizing models of the missions and sensors. Plan execution consists of formatting and submitting requests to missions, continuous monitoring and, if necessary, replanning based on the results of submitting requests and other unexpected events.

[1]For reasons of space, this paper contains an informal discussion of DESOPS capabilities and defers a more formal description of the computational problem and solution algorithms used by DESOPS to future papers [8]

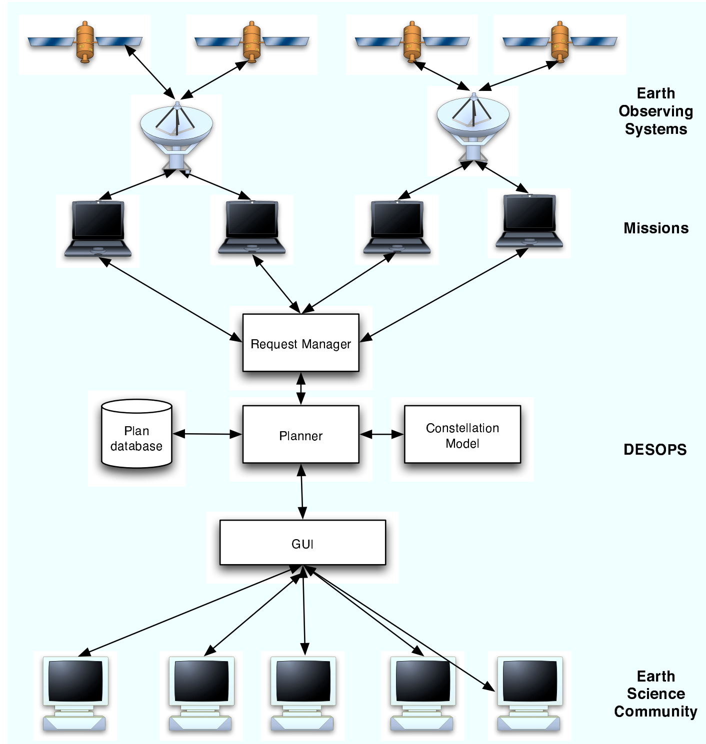

Fig. 1. Architecture for model-based observing

A. Constellation Model

A **constellation model** consists of a database and set of functions for describing the resources available for observation. There are five components to a constellation model:

1) a description of the capabilities of a collection of *sensors*;
2) a model of *time*. For our purposes, time is a finite set of totally ordered values naturally interpreted as the set of days in which some observation can be taken or some other event of interest happens;
3) a model of *geographic space* for identifying the locations and extents of regions to be measured. For example, a region of interest could be specified as a set of latitudes and longitudes to define arbitrary polygons on the Earth;
4) a *satellite orbit* function for determining the set of sensor viewing times for a specified region of interest; and
5) for each resource, a *mission model* that describes constraints on the process by which tasks on the sensor are scheduled by the mission that manages it.

Collectively the constellation model provides a language for specifying the requirements for using a collection of sensing resources.

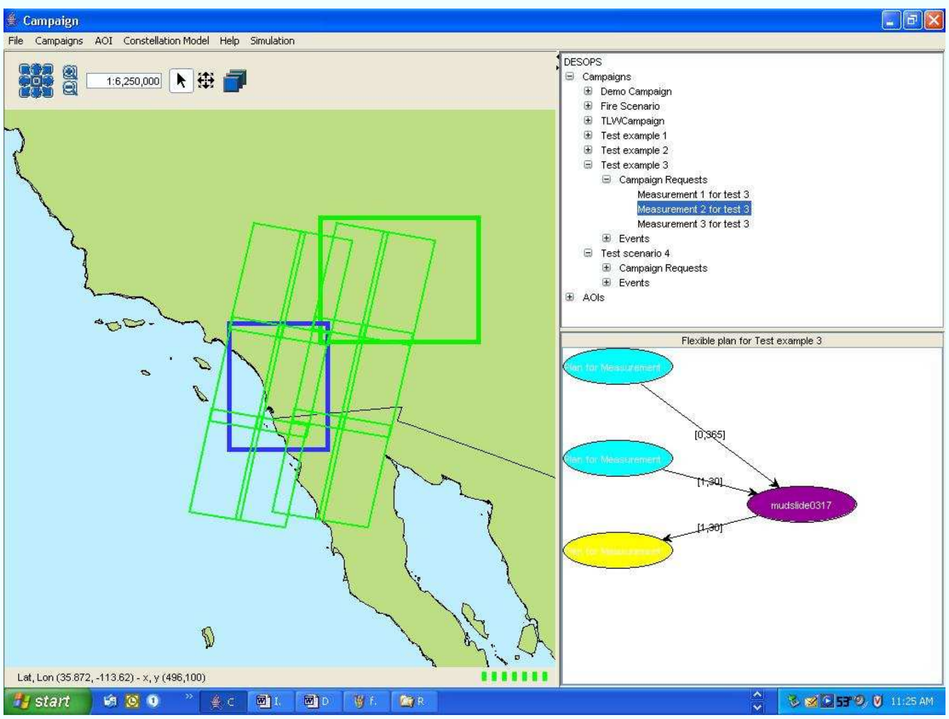

Fig. 2. Graphical user interface for defining coordinated campaigns. The blue box represents the region of interest geographic constraint, green boxes represent view paths that satisfy that constraint. Also shown are a executing flexible campaign plan depicted as a network, and a tree representation of the plan objects and constraints.

B. User Inputs

Users provide inputs through a **graphical user interface**. Users define a campaign to consist of a set of *measurements*, an (optional) set of *exogenous events* (such as a fire or volcano); and a set of *constraints*. Constraints restrict the way a campaign can be carried out. DESOPS supports five kinds of constraints:

1) *sensor constraints* that define a list of sensors through which a measurement to be acquired, with optionally defined preferences for one or more sensors on the list;
2) *temporal constraints*, either in the form of a *time window* (a range of times) for taking a measurement, or ordering restrictions, either between pairs of measurements, or between measurements and exogenous events. In addition, the user may optimally specify *preferences* for time values for these constraints; for example, a user may express a preference for measurements to be ”as close as possible” to others, following the approach in [3].
3) a *geographic constraints* for each measurement, each specified as a set of latitudes/longitudes;
4) a *constraints on data characteristics* for specifying requirements for cloud-free observations, for example, and
5) *cost* constraints.

The main screen of the interface is displayed in Figure 2. This screen shows a map for specifying regions of interest for a campaign, a flexible plan (defined in more detail below) and a textual representation of a campaign as a hierarchy of measurements and constraints. The overpass swaths for one of the requested satellites have also been computed automatically and is visually displayed.

C. Creating Campaign Plans

A *flexible plan* is a concise representation of a set of possible solutions to a campaign scheduling problem. The role of the **Planner** is to build and manage flexible plans. First, the planner constructs an initial flexible plan based on user inputs. Second, new constraints can be added by propagating the effects of an initial set of temporal orderings. In particular, the planner generates start times for each sensor in the domain of each measurement from *view paths* over specified regions of interest during specified time windows. A view path is the intersection of a specified region of interest with the path followed by a satellite over the user-specified time window. In DESOPS, view paths are generated by conducting a web search for these data from mission web sites. Alternatively, it is possible to generate these data directly through the use of simulators such as STK (Satellite Tool Kit). Converging on a flexible plan is an iterative process in which the user is allowed to view and revise the inputs to the problem.

A flexible plan can be represented as a network of nodes representing events or measurements and directed arcs labeled by constraint information. An example is found in Figure 3. The plan consists of three measurements and one event. The constraint $[40, 100]$ represents the belief that event $E1$ is expected to happen sometime between day 40 and day 100 of the campaign. The other constraints represent temporal

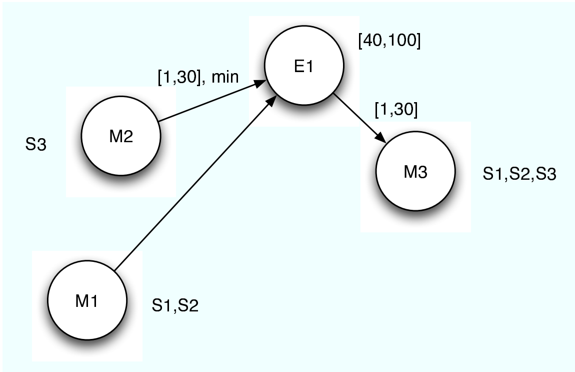

Fig. 3. A simple flexible plan. Each node of the network represents a measurement or event. Directed arcs depict temporal orderings, labeled with the duration between event(s) and measurements(s) The sensors are listed with each measurement.

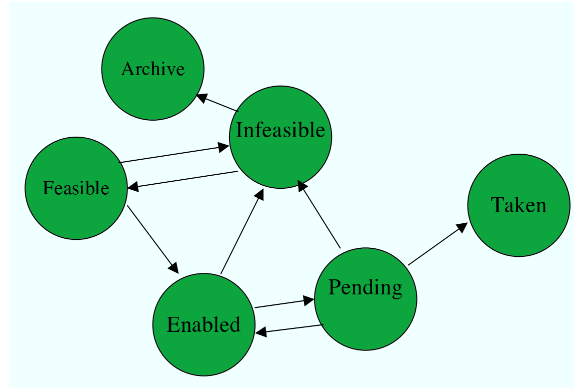

Fig. 4. A state transition model for measurements. States and possible transitions between them are depicted.

ordering constraints; for example, the label between $M1$ and $E1$ expresses the constraint that $M1$ should occur between 1 and 30 days before $E1$, with a preference for times as close to $E1$ as possible. The sensor constraints are also attached to each measurement in the plan. As illustrated in Figure 2, a flexible plan network is part of the visual displays available to the user during campaign definition. The same display is used during campaign execution to provide the user with status information about the plan. For example, in the figure, the blue nodes indicate measurements that have been acquired and the yellow node represents an exogenous event that has yet to occur.

D. Plan Execution

An *observation request* is a specific assignment of a sensor, a time, and a location to the measurement. A *feasible observation schedule* is a sequence of observation requests that satisfy the user specified constraints. In general, a flexible plan gives rise to a number of feasible observation schedules. The **request manager** incrementally executes a feasible observation schedule by submitting observation requests to missions. The request manager also *monitors* the state of the executing plan and initiates *rescheduling* activities where necessary. To carry out these functions the request manager implements an execution strategy for dealing with uncertainty in the execution environment and applies a state- transition model to monitor the progress of the plan.

An *execution strategy* is based on the mission model and information about exogenous events. The mission model advises the request manager on matters related to which mission is most likely to be able to fulfill a request, as well as how and when to submit the request. For example, the mission model may contain a *load profile* for each sensor, which indicates the percentage of time the sensor has been idle during a specified period. The request manager may apply this information by preferring sensors with a smaller load. Second, a mission model contains formatting rules for request submssion. Third, a mission model contains deadlines for submitting requests based on the mission-scheduling process.

The DESOPS *state-transition model* identifies possible states of the overall plan, the component measurements in the plan, and, for each measurement, the state of each associated observation request. The model also defines transitions between states. The request manager implements the plan state-transition model as the mechanism for monitoring the progress of the plan. The request manager observes whether enabling conditions for a transition are met, and, if they are, records the change in state. The state-transition model also allows the request manager to detect when a campaign has failed during execution, which triggers a suspension of the campaign and notification to the user for rescheduling purposes.

Figure 4 shows a state transition model for a measurement. A measurement starts in a feasible state. It becomes enabled when the temporal preconditions for taking the measurement are met (for example, an exogenous event happens or a dependent measurement has been acquired). It becomes infeasible if the constraints make it impossible for it to be taken; this can happen, for example, if all submissions of requests for the measurement are rejected. Otherwise, a measurement is pending if at least one request for the measurement has been submitted. If a mission accepts the request and the image is acquired, the measurement enters the terminal node *Taken*. The user may decide during execution to use data in an archive to acquire the needed data. If so, the request manager no longer submits requests for the observation to the missions.

E. Replanning

As the campaign plan executes, observations or exogenous events happen that can potentially render a campaign plan infeasible.

At this point, the user decides whether to restore feasibility to the plan or to abort it. Plans are made feasible by relaxing constraints that contributed to making the plan infeasible. Figure 5 shows a simple plan that was made infeasible during execution. Exogenous event $E1$ happened at time 69. A constraint requires measurement $M3$, which has yet to occur, happen between 1 and 30 days after $E1$. $M3$ has two observation opportunities: with sensor $S2$ at time 100 or with sensor $S3$ at time 120. Clearly, both exceed the upper bound

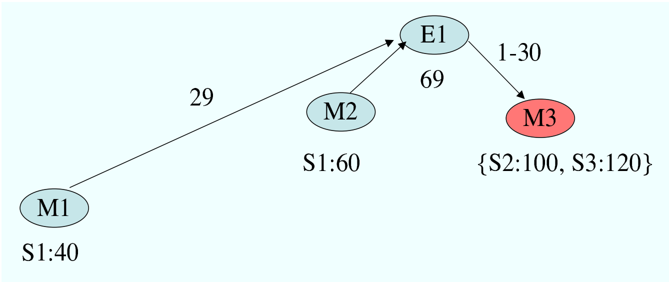

Fig. 5. A replanning scenario. The occurrence of $E1$ at time 69 has made it impossible to schedule an observation of $M3$ that satisfies the constraint between $E1$ and $M3$. The user must decide whether to relax the constraints on the plan to restore its feasibility.

on the temporal ordering constraint, and so this constraint is violated. The user may relax the upper bound of the temporal constraint to make the observation opportunities consistent with the plan. Alternatively, the user may add additional sensors for $M3$ that include opportunities consistent with the ordering constraint, or may decide to acquire $M3$ data through an archive.

DESOPS provides the user continuous plan execution status when requested. It also provides notification of the need for plan repair when the plan becomes infeasible during execution. Visual and textual information will be provided by DESOPS' explanation facility, using a model to map plan state information into useful textual or visual advice.

IV. Implementation and Future Enhancements

The DESOPS system design described in this paper is being implemented in C++ and Java. The implementation is built upon previous work on the AMPS/MOPSS system and the EUROPA constraint-based planning system [7]. An end-to-end prototype with a subset of the capabilities described in this paper is currently being tested and evaluated. Future reports will document the results of these tests.

We distinguish here between two classes of enhancements: local, which expand the capabilities of the current DESOPS described in this paper; and global, which expand DESOPS into a more complete set of capabilities for integrating observing, data analysis and modeling, as described in [1]. Local enhancements for the system include:

1) *Optimization planning*. The current planner does not fully support the application of user-defined utility criteria to generate and execute high-utility campaigns.
2) *Planning under uncertainty*. The current planner does not support the representation of uncertainty associated with exogenous events defined for campaigns. This capability will allow plans to be generated that are optimal with respect to expected behavior of exogenous events, as described in [2].
3) *Middleware services*. The current DESOPS implementation does not support services related to security, login and access, remote process management, storage access and data management.

There are three broad classes of global enhancements that offer the means of expanding DESOPS into a complete set of capabilities for accomplishing the Earth science vision. First, an integration of Earth science domain models into DESOPS would enable the system to advise a user in formulating campaigns. For example, such models could advise users on the selection of promising regions-of-interest for developing a fire campaign. Second, the integration observation scheduling with planning for data analysis as discussed in [6] would lead to an end-to-end system for generating data products. Third, providing the automated means of transforming the results of image analysis into goals for future observation scheduling, as demonstrated on EO-1 [5] would "complete the loop" in automated campaign execution.

V. Conclusion

This paper has described a set of capabilities for building and executing sequences of observations for accomplishing complex campaign goals. Observation requests generated from user inputs describing campaign goals and constraints are submitted electronically to mission operations planners, who then decide whether and how to incorporate the request into future mission schedules. The system also supports dynamic replanning in response to request rejection or unexpected changes in the observing environment. The overall approach to distributed planning has the advantage of allowing missions to maintain ultimate control over their instruments while at the same time allowing Earth scientists more visibility into the resources available for accomplishing their science objectives.

Acknowledgment

Thanks to Jeremy Frank and Joseph Coughlan for comments on previous drafts. Also to Lina Khatib, Christopher Bruno and Andrew Bachman for past contributions of ideas and for their development efforts.

References

[1] *NASA Earth Science Enterprise Strategy*, 2003.
[2] R. Morris et. al., *Temporal Planning with Preferences and Probabilities*, Proceedings of the ICAPPS Workshop on Constraint Programming for Planning and Scheduling, 2005.
[3] P. Morris et. al, *Strategies for Global Optimization of Temporal Preferences*, Proceedings of CP 2004: 408-422.
[4] B. Clement and A. Barrett, *Coordination Challenges for Autonomous Spacecraft*, AAMAS-02 Workshop Notes on Toward an Application Science: MAS Problem Spaces and Their Implications to Achieving Globally Coherent Behavior, 2002.
[5] S. Chien et. al., *Autonomous Science on the EO-1 Mission*, International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2003). Nara, Japan. May 2003.
[6] K. Golden et. al., *Automating the processing of Earth observation data*, Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation for Space (i-SAIRAS), 2003.
[7] D. Smith, J. Frank, and A. Jonsson. *Bridging the Gap Between Planning and Scheduling*, Knowledge Engineering Review, 15:1, 2000.
[8] R. Morris et. al., *An Architecture and System for Scheduling Coordinated Earth Science Measurements*, Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation, Munich Germany, 2005.
[9] R. Buyya, Abramson, D. and Venugopal, S., *The Grid Economy*, Special Issue on Grid Computing, Proceedings of the IEEE, Manish Parashar and Craig Lee (editors), Volume 93, Issue 3, 698-714pp, IEEE Press, New York, USA, March 2005.